



# RFC 6550: On Minimizing the Control Plane Traffic of RPL-based Industrial Networks

Dimitrios Sourailidis, Remous-Aris Koutsiamanis, Georgios Papadopoulos,  
Dominique Barthel, Nicolas Montavont

## ► To cite this version:

Dimitrios Sourailidis, Remous-Aris Koutsiamanis, Georgios Papadopoulos, Dominique Barthel, Nicolas Montavont. RFC 6550: On Minimizing the Control Plane Traffic of RPL-based Industrial Networks. WoWMoM 2020 - IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks", Aug 2020, Cork, Ireland. pp.439-444, 10.1109/WoWMoM49955.2020.00080 . hal-02749045

**HAL Id: hal-02749045**

**<https://hal.science/hal-02749045>**

Submitted on 3 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# RFC 6550: On Minimizing the Control Plane Traffic of RPL-based Industrial Networks

Dimitrios Sourailidis<sup>†</sup>, Remous-Aris Koutsiamanis<sup>†</sup>, Georgios Z. Papadopoulos<sup>†</sup>,  
Dominique Barthel<sup>‡</sup> and Nicolas Montavont<sup>†</sup>

<sup>†</sup> IMT Atlantique, Irista, France, Email: {firstname.lastname}@imt-atlantique.fr

<sup>‡</sup> Orange Labs, France, Email: dominique.barthel@orange.com

**Abstract**—The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is the de facto routing protocol for Low Power and Lossy Networks (LLNs). It is a proactive and link-layer agnostic routing protocol standardized as RFC 6550 by the Internet Engineering Task Force (IETF). Based on the distance-vector technique, RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) topology. To establish and maintain the routes, RPL uses DODAG Information Object (DIO) control packets, that are transmitted in broadcast, for RPL nodes to propagate the DODAG related information, while its transmission frequency depends on Trickle timer algorithm, i.e., the less stable the network the more DIOs are transmitted. Thus, when a new node intends to join the RPL DODAG, it listens for a DIO message from nearby nodes, which may take a very long time if the network is in a stable state. Therefore, RFC 6550 is equipped with the DODAG Informational Solicitation (DIS) message to solicit DIOs from nearby RPL nodes, similar to the Router Solicitation in IPv6 Neighbor Discovery. However, the solicitation procedure is not the most efficient one, since it resets the Trickle timers in the nodes that receive the DIS message and, thus, they transmit an unnecessarily large number of DIOs that congest the network and consume energy in the nodes. In this paper, we propose to augment RFC 6550, the RPL routing protocol, with additional DIS flags and options that allow a RPL node to better control how the nearby RPL nodes will respond to its solicitation for DIOs. Our performance evaluation in Contiki-NG & COOJA demonstrates that we can reduce the control packets in the network by up to 45.5%.

**Index Terms**—Internet of Things, IoT, Industrial IoT, IIoT, Industry 4.0, LLNs, RPL, RFC 6550, Control Plane, DIS, DIO

## I. INTRODUCTION

Industry 4.0 is an emerging domain in the Internet of Things (IoT) [1]. It aims at ultra reliable wireless communication for secure and adaptive automation of production chains. Indeed, in scenarios such as smart meters or factory automation, for instance in vehicle production, Low Power and Lossy Networks (LLNs) are required that consist of thousands of nodes<sup>1</sup> that are equipped with radio interface(s) and with multiple sensors and actuators [2]. Additionally, in order to extend the industrial network beyond the radio coverage of one node, a mesh technology is required to allow some nodes to act as a relay for others. However, beyond one hop, a routing protocol is also required to route the packets over the network.

The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [3] is one of the most adopted routing

protocols for LLNs [4], and the focus of this work. RPL is a proactive routing protocol, with the benefit of actively maintaining connectivity, but it requires control traffic to maintain the network topology up-to-date and available. The information that allows nodes to connect to the network to update their routing knowledge is broadcast periodically (but with a varying period) by already connected nodes. It can also be requested by joining nodes to speed up the connection process by avoiding the wait until the next periodic broadcast. RPL performs an important optimization of the period of this broadcast: the more stable the topology of the network, i.e., the lower the variability of reliability, the less frequent the broadcasts. The intuition is that things change slowly in a stable network, and thus energy can be conserved by sending fewer updates. Conversely, when higher variability is detected, the period is shortened to provide faster updates to keep up with the changes, at the cost of additional energy consumption.

In this work, we first investigate the problem of inappropriate lowering of these broadcast intervals when connection information requests are received. Secondly, we address the related issue of high network traffic overhead when the connection information requests are made, which is created by multiple receivers trying to reply to the requests at the same time. To solve these issues, we propose an augmentation of RFC 6550, the RPL routing protocol specification. We propose adding three new flags to the connection information request message: the “No Inconsistency” flag so that nodes that receive the requests do not lower their broadcast intervals, the “Type” flag to specify the type (unicast or multicast) of response, and the “Option Request” flag to allow requesting specific partial connection information to be provided in the response, instead of everything available. Moreover, we propose two new fields, one to allow a way to restrict the set of responding nodes, and another to request that responders spread out their responses in time to reduce transmission collisions. We perform a thorough performance evaluation with COOJA, a simulator for the Contiki-NG operating system [5]. Our results demonstrate that, by including the additional flags and options inside the request control packet, the number of control packets sent by nearby nodes in response to the request is 0.58 times the number of packets without our changes and the number of control packets received as a response to the request are 0.54 times the number of packets without our changes.

<sup>1</sup>“Node”, “device”, “mote”, and “RPL router” are used interchangeably.

The rest of the paper is organized as follows. In Section II, we provide a background of RFC 6550, and we present the problem statement. In Section III, we present the enhancements to the connection information request control packet with the proposed additional flags and options. In Section IV, we describe the simulation setup, and demonstrate the attained performance of our proposed approach. Finally, Section V provides the concluding remarks as well as our future work.

## II. TECHNICAL BACKGROUND : RFC 6550

### A. RPL

RPL is a proactive routing protocol produced by the Routing Over Low power and Lossy networks (ROLL) Internet Engineering Task Force (IETF) Working Group (WG). RPL builds a Destination Oriented Directed Acyclic Graph (DODAG) by employing a distance-vector technique. In each DODAG, there is a single node considered to be the “root” which serves as the gateway to other non-RPL networks. Based on a common objective function [6], [7], [8] each node selects one or more parent(s), acting as a relay toward the root node. In such a network, the traffic from non-root nodes towards the root is called the upstream traffic, while the reverse is called the downstream traffic.

In order to form, maintain and update a DODAG, the RPL nodes periodically transmit, in broadcast, DODAG Information Object (DIO) control packets. A DIO message carries DODAG related information, such as RPL Instance, DODAGID, DODAGVersionNumber, or routing related metrics that will allow a node to select its parent. The interval of the DIO transmissions depends on the network stability. Indeed, the more stable the network, the fewer the DIO packets sent in order to reduce the overhead. Correspondingly, when the network is not stable, more DIO messages are transmitted. This timing algorithm is called a Trickle timer [9].

### B. The Trickle Algorithm

RPL nodes transmit DIOs using a Trickle timer [9]. The Trickle algorithm increases its interval or stays stable only if it detects transmissions that are “consistent”. An example of an event which is considered an inconsistency and which leads to a Trickle timer reset is the reception of a DIO with a new DODAGVersionNumber.

### C. Problem Statement

Indeed, when a new device intends to join a DODAG, it turns its radio *ON* and “waits” to receive a DIO. This is a typical use case from smart grid applications, when a smart meter is being replaced in the field, while a RPL network is already operating [10], [11]. However, if the DODAG is in a stable state, then the joining node will have to wait for a very long time. RFC 6550 supports a DODAG Information Solicitation (DIS) control packet to optimize the joining procedure. Thus, a RPL node may employ a DIS message to solicit DIO messages from nearby RPL routers. However, since the new node does not have knowledge of the operating network, e.g. which neighbor routers it should query, it solicits DIOs responses using a multicast DIS. As per RFC 6550 standard, when receiving

Table I: Behavior of a RPL router according to RFC 6550 when it receives a DIS control packet. SI Option refers to the presence of this option in the DIS, Match refers to whether the SI predicates match, UC stands for Unicast, and MC stands for Multicast.

DIS request			Receiver behavior	
SI option	Match	UC/MC DIS	Send UC/MC DIO	Reset Trickle Timer
×	-	UC	✓ UC	×
×	-	MC	✓ MC	✓
✓	×	UC	×	×
✓	×	MC	×	×
✓	✓	UC	✓ UC	×
✓	✓	MC	✓ MC	✓

0	7 8	15 16
Flags	Reserved	Option(s)...

Figure 1: DIS Base Object.

a multicast DIS, the neighbor routers will *reset their Trickle timers*, and will immediately transmit their DIOs responses. The disadvantage of resetting the Trickle timers is that the nodes will start transmitting their periodic DIOs at the shortest time interval for a considerable duration until the Trickle timers reach again the “relaxed” time intervals. This procedure will introduce unnecessary control traffic in the network that will increase congestion, and consequently will consume energy on the nodes, a problem on potentially battery-operated devices.

A node in RPL can send DIS messages in either multicast (to all neighbor routers) or in unicast (to a specific neighbor). When doing this, there is some ability in the RFC 6550 standard to control the form and contents of the DIO response, but it is limited. More specifically, a DIS can have a Solicited Information (SI) option that specifies which predicates of the DODAG should be matched. If no SI option is present then all the available information is reported in the response DIO. The required behavior of the DIS receiver is summarized in Table I. It is noteworthy that the reception of multicast DIS with no SI option or with a matching SI option resets the Trickle timer.

Therefore, as previously described, when a new node intends to connect and it is not aware of which node to send a unicast DIS to, it needs to send a multicast DIS. Every time this happens, all the receiving nodes reset their Trickle timer.

## III. DIS MODIFICATIONS

### A. DIS Base Object

In this section, we present the three flags that we propose to be included in the DIS object base, i.e., inside the 8-bit unused field reserved for flags. In Figs. 1 and 2, the default from the RFC 6550, and the proposed DIS Base Objects with the three flags, *i*) “NO Inconsistency” (N), *ii*) “DIO Type” (T), *iii*) “DIO Option Request” (R), are depicted.

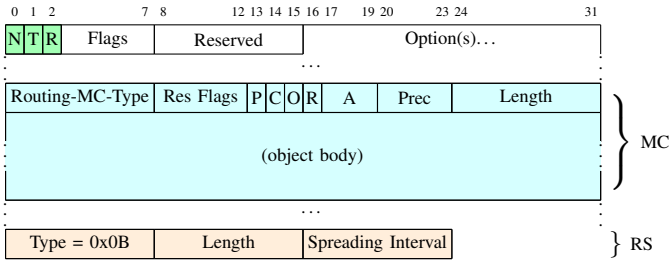


Figure 2: New DIS Base Object with the N, T and R flags in the DIS Base Object and examples of the MC and RS Options.

1) “NO Inconsistency” (N) Flag: when a RPL router receives a multicast DIS control packet with the N flag set, it will not reset the Trickle timer. However, it must take an action, transmitting a response DIO explicitly. A *configuration option* must be included in this DIO. Indeed, this operation augments RFC 6550 [3], which had provision for such flag. Considering that this DIO transmission is not a consequence of the Trickle timer algorithm, it will be sent right away if no *Response Spreading option* is included (see in the following section about the Response Spreading option).

2) “DIO Type” (T) Flag: the T flag specifies what type of DIO is sent in response, when the N flag is set. Thus, when a multicast DIS control packet with the T flag set is received, a RPL router receiving it must respond with a unicast DIO, and it must respond with a multicast DIO if this flag is unset.

3) “DIO Option Request” (R) Flag: when a RPL router receives a DIS control packet with the R flag set, in its DIO response, it must include all the options that were requested by the DIS containing one or multiple DIO Option Request options. Indeed, it must not include DIO options that were not explicitly requested. This operation contradicts RFC 6550 [3] which requires including a Configuration option in all DIOs requested by a unicast DIS.

When a unicast DIS is transmitted, both its N and T flags should be set to 0, i.e., set to the default values of RFC 6550 [3]. Thus, when a RPL router receives a *unicast* DIS control packet, the N and T flags will be ignored.

## B. DIS Options

Furthermore, we propose two options, *i)* the DIS Metric Container (MC) option, and *ii)* the Response Spreading (RS) option, in order for a RPL node to better control the DIOs responses to DIS queries from nearby nodes.

1) *DIS MC*: In order to reduce the number of nodes that will respond to a DIS control packet, we propose to include routing constraints in the DIS. Thus, only the node(s) that fulfill these constraints can respond to the DIS. In this paper, we propose to augment the RFC 6550 standard [3] by including an MC option [12] inside the DIS messages. Thus, the routing constraints can be described by employing the MC option that is contained in the DIS, see Fig. 2.

Then, the RPL router is allowed to respond to the DIS message only if it satisfies the constraints. It should be noted

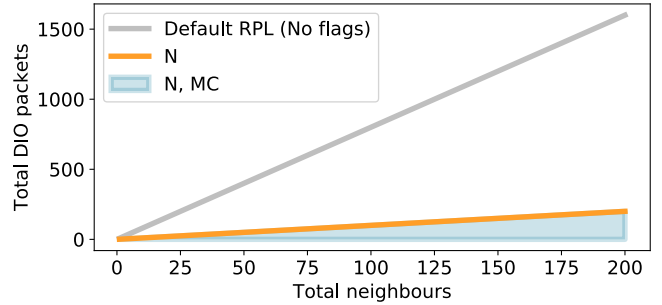


Figure 3: Theoretical count of DIO packets sent in response to one DIS packet as a function of the number of neighbors receiving the DIS packet and the DIS flags and options used. The shaded region expresses the range of DIO counts depending on how many ( $[0 \dots M]$ ) of the  $M$  neighbors are filtered out by the use of the metric in the MC option.

that this option can be used in both unicast and multicast DIS transmissions.

Comparing the theoretical performance of the default RPL (no flags) case with the N flag and the MC option, as shown in Fig. 3, the effect of the N flag is to significantly reduce the number of DIOs sent. The MC option further reduces their number. It is important that care is taken when using the MC option, because too restrictive constraints might lead to all available neighbors being disqualified from answering.

2) *Response Spreading*: Even by considering Metric Container option inside the DIS control packet, a multicast DIS transmission may still introduce a large number of immediate DIO responses from the nodes. However, it is well-known that concurrent transmissions from multiple nodes may lead to repeated collisions and, thus, packet losses. In order to overcome such issues, in this paper, we define an optional DIO *Response Spreading* mechanism, as depicted in Fig. 2, with a recommended Type value 0x0B.

When a RPL router receives a DIS message that includes the Response Spreading option, before attempting to send its DIO, it must wait first for a time uniformly chosen in the interval  $[0, 2^{SpreadingInterval}]$  that is expressed in *ms*.

Note that the Response Spreading option should not affect the Trickle timer algorithm. Moreover, this option may be included inside a DIS control packet that is transmitted in unicast, however there is no benefit in applying it. Finally, multiple Response Spreading options should not be included in a single DIS control packet.

## IV. PERFORMANCE EVALUATION

### A. Simulation Setup

Our simulation environment involves 10 fixed RPL nodes (including the DODAG root) in the topology shown in Fig. 4. Each node transmits UDP data packets at constant rate of 1 packet per 50 minutes ( $\pm 1$  sec to create a slight variability), with a payload size of 8 bytes so no fragmentation is required. In this

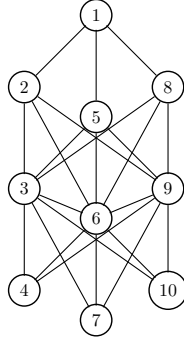


Figure 4: Network topology.

work, we are evaluating the effect of DIS parameters on the transmission of DIO control packets, therefore the UDP traffic rate is set at a low level to minimize the interference with control packet measurements. To study the effect of the augmented protocol with the additional flags and options we have used a scenario where a node (i.e., node 6) enables and disables its radio every 30 minutes (30 min kept enabled, 30 min kept disabled, starting in the disabled state). This allows us to examine the behavior of a node when it (re)connects to a RPL network. The simulator makes it more complicated to simulate a RPL router device power-on or power-off, so we use the enabling and disabling of the radio instead. The 30 min interval is important because it is bigger than the largest Trickle timer value (1048 sec) so the rest of the nodes which do not enable and disable their radios will have stabilized and reached the maximum Trickle timer interval value. This reflects a stable network, so when node 6 tries to (re)connect, we can examine the disturbance it creates on the rest of the stable nodes (e.g., Trickle timers reset).

Furthermore, we employed the IEEE Std 802.15.4-2015 Timeslotted Channel Hopping (TSCH) [13] Medium Access Control (MAC) layer with the 6TiSCH Minimal [14] Scheduling Function, and the RPL routing protocol. The details of the simulation setup are presented in Table II.

Finally, all the proposed flags (N, T, R) and options (MC & RS) have been implemented in Contiki-NG, and all the simulations were done with the COOJA simulator. However, for the purposes of investigating the effect of the DIS options on DIO control packets, we only investigate two flags (N & T) since the R flag has no effect on the timing and the count of DIO packet transmissions, only on their contents. In all cases, when a node sends a DIS to connect to the network, it is a multicast DIS.

In the next sections, we present the results of the simulation which compare the performance of the augmented RFC 6550 against the standard version.

### B. Simulation Results

We present the results from one of the replications in Fig. 6, in the form of a timeline of the events during a full 180 minute simulation of the network.

For the default RPL implementation (i.e., no flags), Fig. 6a shows that every time node 6 requests to join the network via

Table II: Simulation parameters.

Simulation	
Operating System	Contiki-NG
Simulator	COOJA
Duration	180 min
Data traffic rate	1 pkt / (3000±1 sec)
Replications / RNG Seeds	10
Topology (see Fig. 4)	
Topology Type	Multi-hop
Link quality	100%
Number of nodes	10
PHY & MAC layer	
Protocol	IEEE Std 802.15.4-2015 TSCH
Scheduling Function	6TiSCH Minimal
Enhanced Beacon (EB) period	16 sec
Timeslot length	10 ms
Slotframe length	3 Timeslots
Nº of channels	4
Nº of RTX	7 RTX
Routing layer	
Routing protocol	RPL
DIS type	Multicast
Minimum DIO interval ( $I_{min}$ )	4 sec
Maximum DIO interval ( $I_{max}$ )	1048 sec
Nº of UDP sources	9 (nodes 2-10)
Nº of UDP sinks	1 (node 1)
UDP payload size	8 bytes
DIS flags and options	
Flags	
N	T
×	-
✓	×
✓	×
✓	×
✓	×
✓	✓
✓	✓
✓	✓
✓	✓
✓	✓
✓	✓
Options	
MC	RS
-	-
×	×
✓	×
×	✓
✓	✓
×	×
✓	×
×	×
×	✓
✓	✓

multicast DIS after a period of disconnection, all the nodes receiving the packet are triggered and have their Trickle timers reset and as a result they start sending frequent DIOs. In contrast, when the nodes receive a multicast DIS with the N flag set, as shown in Fig. 6b, they do not reset their Trickle timers and they immediately respond with a DIO, therefore reducing the total multicast DIOs sent. Similarly, when both N & T flags are set, as shown in Fig. 6c, the number of DIOs is further decreased since the DIOs are now also unicast and as a result suffer less interference. Adding the MC option to both N only (see Fig. 6d), and N & T (see Fig. 6e) cases, further reduces the DIOs, since a subset of neighbors now respond, however, the reduction is less dramatic than the introduction of the N flag and more prominent in the N only case, since there is more room for improvement there. A similar result is produced if, instead of the MC option, we add the RS option to both N only (see Fig. 6f), and N & T (see Fig. 6g) cases. Finally, when both the MC and RS options are added to both

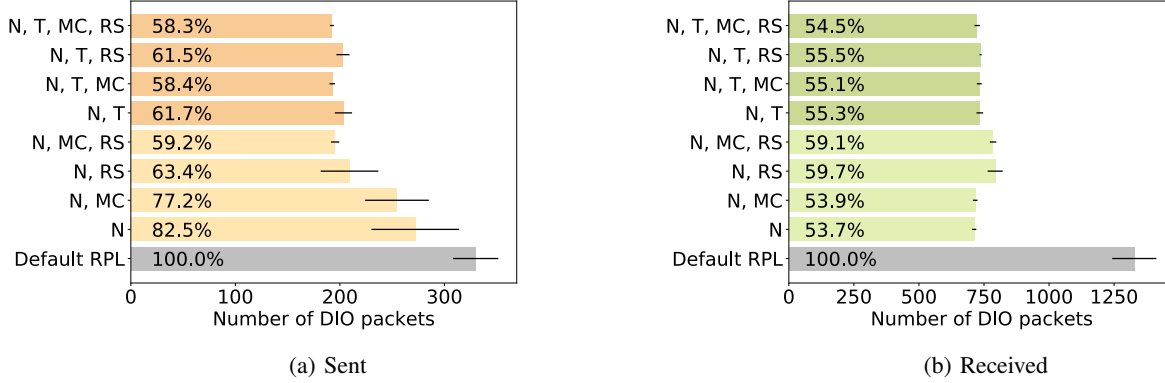


Figure 5: Mean and standard deviation of the total number of DIO packets sent (a) and received (b) in the network with the default RPL implementation and with all the valid combinations of the proposed DIS flags and options. Additionally, the DIO packets sent (a) and received (b) in each case as a percentage of the default RPL case is shown in the bars for ease of comparison.

N only (see Fig. 6h), and N & T (see Fig. 6i) cases, it becomes clear that these are the best performing cases. It is noteworthy that in some cases, here in Figs. 6g & 6i, it can be seen at approximately the 150 minute mark, right after the third radio re-enabling, that no unicast DIO is received. This can happen because, as a result of the Trickle algorithm, DIOs are broadcast every so often irrespective of the DIS received. Therefore, it can happen that node 6 receives a DIO before sending its DIS, thus connecting without explicitly asking for a DIO.

A summarized view of the total control traffic overhead is shown in Fig. 5, with the total number of DIO sent and received in the network shown in Figs. 5a & 5b correspondingly. It is noteworthy that the default RPL implementation, in addition to creating and receiving more DIOs, also has high variance, which is not as high in the “N & T” cases. The results show an overall reduction in the number of sent DIOs: 0.58 times the number of DIOs sent with the default RPL. Additionally, the number of received DIOs is also lower: 0.54 times the number of DIOs received with the default RPL. Therefore, the “N, T, MC, RS” case is the best choice when a node tries to connect to a network.

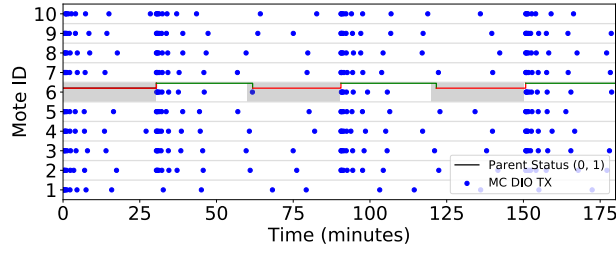
## V. CONCLUSIONS

In this work, we presented the problems of network traffic overhead created when a node joins a stable RPL network. We investigated the problem of resetting of the RPL Trickle timer when multicast DIS control packets are received and the secondary problem of high traffic when multiple DIS receivers all try to send their DIO responses at the same time. To solve these issues, we proposed an augmentation of RFC 6550, in the form of three new flags and two options for the DIS control packet and associated logic in the handling of these flags and options. We implemented and evaluated our proposals in the Contiki-NG OS and the COOJA simulator proving that the implementation is realistic and feasible. The results show very significant performance improvements in terms of reduced overhead. The next step is to expand the testing to much

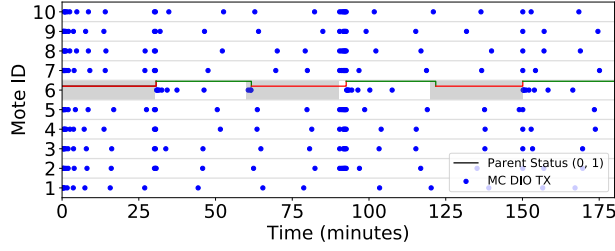
bigger topologies and to measure the impact of this proposal with different traffic and node (dis)connection scenarios.

## REFERENCES

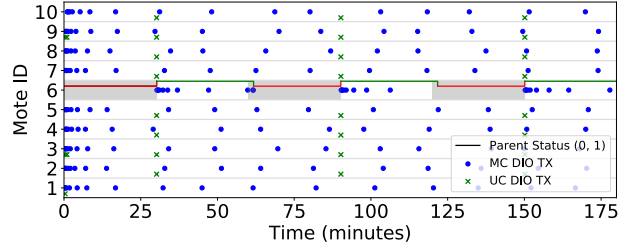
- [1] G. Z. Papadopoulos, P. Thubert, F. Theoleyre, and C. Bernardos, “RAW use cases,” IETF, RAW WG, Internet-Draft draft-bernardos-raw-use-cases-03 [work-in-progress], March 2020.
- [2] K. Pister, P. Thubert, S. Dwars, and T. Phinney, “Industrial Routing Requirements in Low-Power and Lossy Networks,” IETF, RFC 5673, 2009.
- [3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and A. R., “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” IETF RFC 6550, 2012.
- [4] F. Lemercier, G. Habault, G. Z. Papadopoulos, P. Maille, P. Chatzimisios, and N. Montavont, “Communication Architectures and Technologies for Advanced Smart Grid Services,” in *Transportation and Power Grid in Smart Cities: Communication Networks and Services*, M. H. R. Hussein T. Mouftah, Melike Erol-Kantarci, Ed. Wiley, 2018, ch. 8, pp. 217–246.
- [5] “Contiki-NG,” [Online]. Available: <https://www.contiki-ng.org/>
- [6] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” RFC 6719 (Proposed Standard), Internet Engineering Task Force, Sep. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6719.txt>
- [7] C. Ji, R.-A. Koutsiamanis, N. Montavont, P. Chatzimisios, D. Dujovne, and G. Z. Papadopoulos, “TAOF: Traffic Aware Objective Function for RPL-based Networks,” in *Proceedings of the Global Information Infrastructure and Networking Symposium (GIIS)*, 2018.
- [8] R. Koutsiamanis, G. Z. Papadopoulos, E. I. Sanchez, C. Ji, D. Dujovne, and N. Montavont, “Traffic-aware Objective Function,” IETF, ROLL WG, Internet-Draft draft-koutsiamanis-roll-traffic-aware-of-00 [work-in-progress], March 2019.
- [9] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, “The Trickle Algorithm,” IETF RFC 6206, March 2011.
- [10] G. Z. Papadopoulos, “Use cases for DIS Modifications,” IETF, ROLL WG, Internet-Draft draft-papadopoulos-roll-dis-mods-use-cases-00 [work-in-progress], March 2020.
- [11] G. Habault, M. LeFrancis, F. Lemercier, N. Montavont, P. Chatzimisios, and G. Z. Papadopoulos, “Monitoring Traffic Optimization in Smart Grid,” *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 3246–3255, 2017.
- [12] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, “Routing metrics used for path calculation in low-power and lossy networks,” IETF RFC 6551, March 2012.
- [13] “IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs),” IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011), April 2016.
- [14] X. Vilajosana, K. Pister, and T. Watteyne, “Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration,” IETF RFC 8180, May 2017.



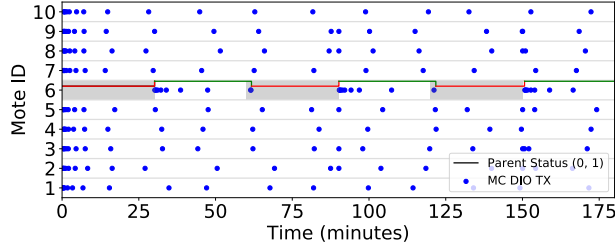
(a) No flags enabled (i.e., default RPL)



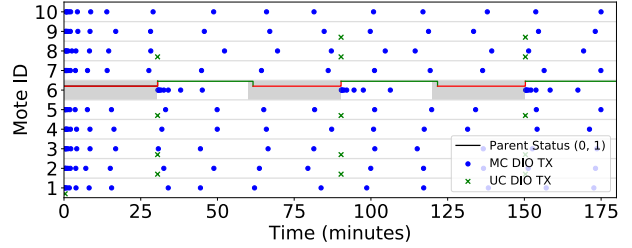
(b) N flag enabled to avoid the Trickle timer reset



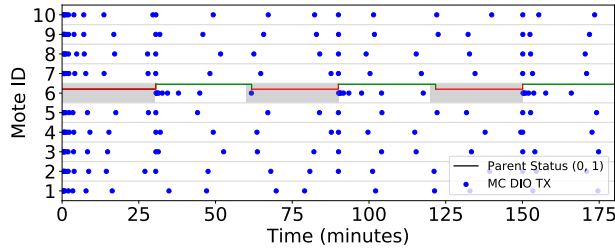
(c) N flag enabled and the T flag to force a unicast response



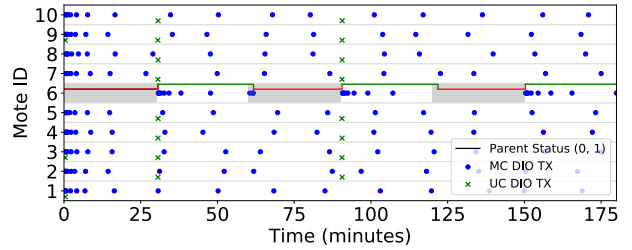
(d) N flag enabled and the MC option to limit responders



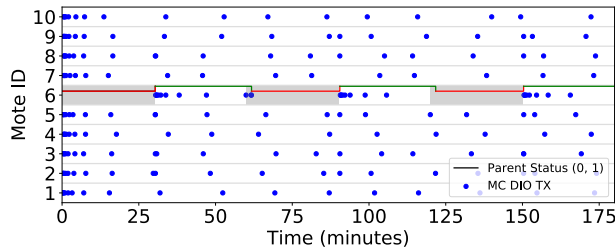
(e) N & T flags enabled and MC option to limit responders



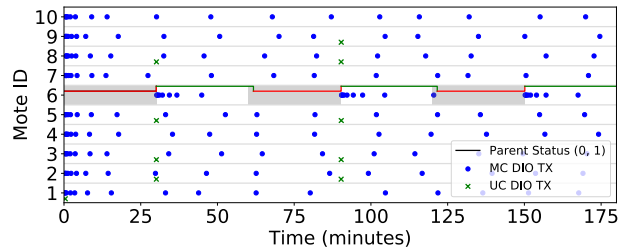
(f) N flag enabled and the RS option for response spreading



(g) N & T flags enabled and the RS option for response spreading



(h) N flag enabled and MC & RS options



(i) N & T flags enabled and MC & RS options

Figure 6: Timeline of MC (Multicast) and UC (Unicast) DIO control packets sent. The periods of time when the radio module of node 6 is disabled to simulate the node turning off is shaded gray. The periods of time when node 6 has joined the DODAG are indicated with a green line, whereas when it does not have one, with a red line.